G. Bagliesi
30/6/98

First implementation of AlephObjects in ALPHA++

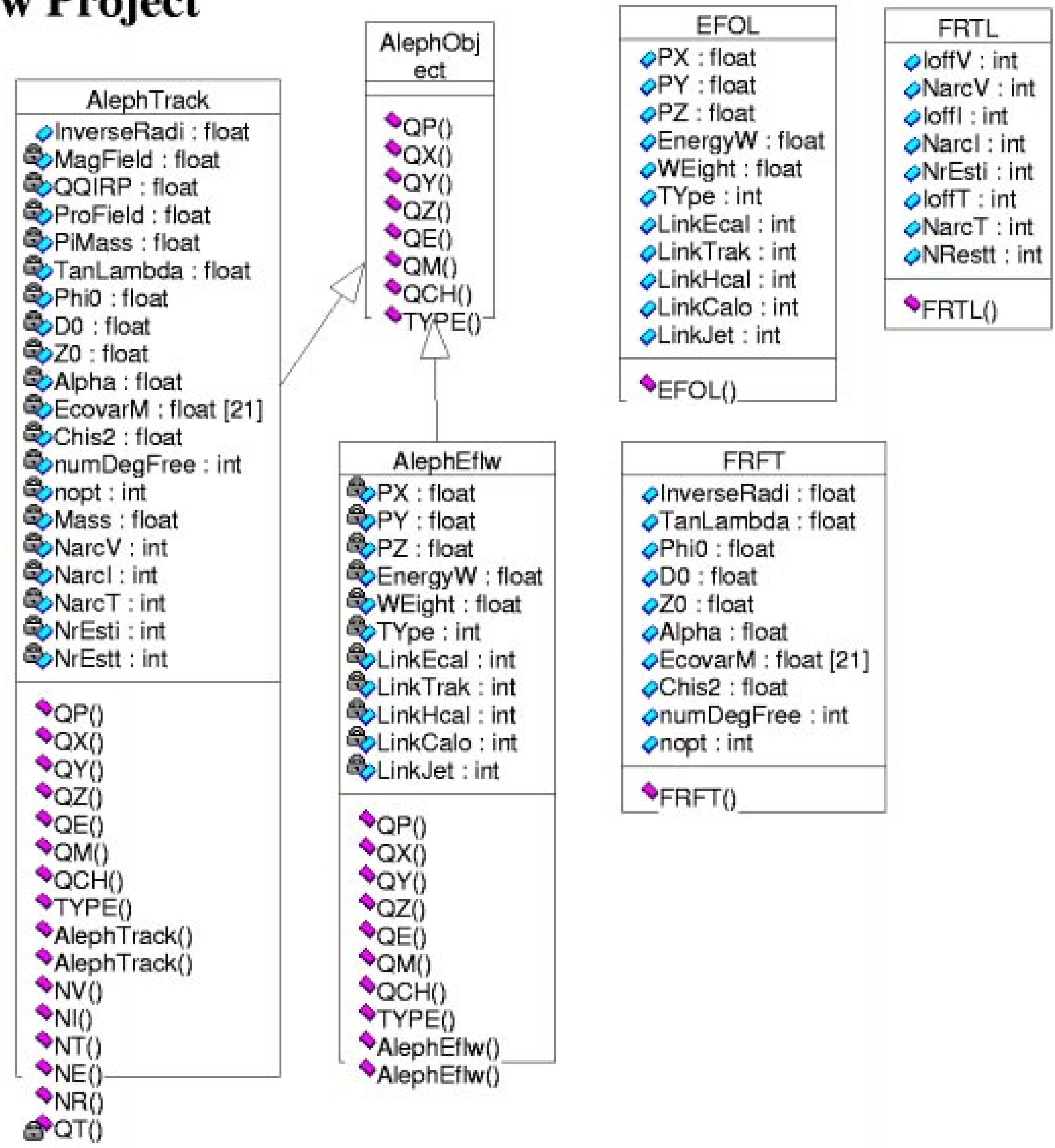
Reproduce the ALPHA structure

- Abstract Class

```
class AlephObject {
 public:
  //  ~AlephObject();
  virtual float QP() = 0;
  virtual float QX() = 0;
  virtual float QY() = 0;
  virtual float QZ() = 0;
  virtual float QE() = 0;
  virtual float QM() = 0;
  virtual float QCH() = 0;
  enum {OBJECT,TRACK,EFLOW,ECAL,HCAL};
  virtual int TYPE(){return OBJECT;};

};
```

New Project



AlephObject
QP()
QX()
QY()
QZ()
QE()
QM()
QCH()
TYPE()
EFOL
PX : float
PY : float
PZ : float
EnergyW : float
WEight : float
TYpe : int
LinkEcal : int
LinkTrak : int
LinkHcal : int
LinkCalo : int
LinkJet : int
EFOL()
FRTL
IoffV : int
NarcV : int
IoffI : int
NarcI : int
NrEsti : int
IoffT : int
NarcT : int
NRestt : int
FRTL()
AlephTrack
InverseRadi : float
MagField : float
QQIRP : float
ProField : float
PiMass : float
TanLambda : float
Phi0 : float
D0 : float
Z0 : float
Alpha : float
EcovarM : float [21]
Chis2 : float
numDegFree : int
nopt : int
Mass : float
NarcV : int
NarcI : int
NarcT : int
NrEsti : int
NrEstt : int
QP()
QX()
QY()
QZ()
QE()
QM()
QCH()
TYPE()
AlephTrack()
AlephTrack()
NV()
NI()
NT()
NE()
NR()
QT()
AlephEflw
PX : float
PY : float
PZ : float
EnergyW : float
WEight : float
TYpe : int
LinkEcal : int
LinkTrak : int
LinkHcal : int
LinkCalo : int
LinkJet : int
QP()
QX()
QY()
QZ()
QE()
QM()
QCH()
TYPE()
AlephEflw()
AlephEflw()
FRFT
InverseRadi : float
TanLambda : float
Phi0 : float
D0 : float
Z0 : float
Alpha : float
EcovarM : float [21]
Chis2 : float
numDegFree : int
nopt : int
FRFT()

AlephTrack.h

```
class AlephTrack : public AlephObject {
 public:
  AlephTrack(FRFT& frft, FRTL& frtl);
  AlephTrack() {};
  // FRFT Related Member functions
  int TYPE();
  float QP();
  float QX();
  float QY();
  float QZ();
  float QE();
  float QM();
  float QCH();
  float InverseRadi;

  // FRTL Related member functions


  float NV();
  float NI();
  float NT();
  float NE();  // ITC following spirals
  float NR();  // TCP following spirals
private:
  // FRFT related data members

  float MagField;
  float QQIRP;
  float ProField;
  float PiMass;
  float QT();
  float TanLambda;
  float Phi0;
  float D0;
  float Z0;
  float Alpha;
  float EcovarM[21];
  float Chis2;
  int numDegFree;
  int nopt;
  float Mass;

  // FRTL related data members

  int NarcV;
  int NarcI;
  int NarcT;
  int NrEsti;
  int NrEstt;
};
```

AlephEflw.h

```
class AlephEflw: public
     AlephObject {
 public:
  AlephEflw(EFOL& efol);
  AlephEflw() {};

  // Mandatory Member
     functions
  int TYPE();
  float QP();
  float QX();
  float QY();
  float QZ();
  float QE();
  float QM();
  float QCH();
private:
  float PX;
  float PY;
  float PZ;
  float EnergyW;
  float WEight;
  int TYpe;
  int LinkEcal;
  int LinkTrak;
  int LinkHcal;
  int LinkCalo;
  int LinkJet;
 };
```

NonPersistentBanks.h

```
class NonPersistentBanks : public HepDbApplication {
 public:
  NonPersistentBanks();
 ~NonPersistentBanks();

 void SetPointer(ooHandle(AlephBank)& bankH);
 void CreateBanks();
 AlephTrack* track;
 AlephEflw* eflow;
 AlephObject** object;
 int Ntracks;
 int Nobjects;
 int Neflow;

private:
 int Nfrft;
 ooHandle(FRFT_Bank) frftH;
 int Nfrtl;
 ooHandle(FRTL_Bank) frtlH;
 int Nefol;
 ooHandle(EFOL_Bank) efolH;
};
```

How to use ...

```
void readApp::TestBosBanks(NonPersistentBanks& bb) {
  int Ntracks = bb.Ntracks;
  cout << " Tracks All " << Ntracks << endl ;
  if(Ntracks>0)
    {
      for(int iii=0;iii<Ntracks;iii++){
       cout << " Momentum " << bb.track[iii].QP() << endl;
       cout << " Nvdet " << bb.track[iii].NV() << endl;
       cout << " Nitc " <<  bb.track[iii].NI() << endl;
       cout << " Ntpc " <<  bb.track[iii].NT() << endl;
}
    }

int Nobjects = bb.Nobjects;
cout << " Objects All " << Nobjects << endl ;
if(Nobjects>0)
    {
      for(int iii=0;iii<Nobjects;iii++){
       cout << " Obj Type " << bb.object[iii]->TYPE() << endl;
       cout << " Momentum " << bb.object[iii]->QP() << endl;
       cout << " Energy " << bb.object[iii]->QE() << endl;
      }
    }
}
```

Problems

- STL not supported by CC
  - Actually NonPersistentBanks implemented with the “new” operator
- ALPHA does **many things**:
  - *Year dependent* treatment of:
    - FRFT, FRTL, EFOL
    - FRFT (2,3,10,12): how to use?
  - Links:
    - EFOL <> TRACKS
    - EFOL <> CALO
  - MINI, NANO *(do we support them ?)*
- What should be taken from the ALPHA internal banks (QVEC etc...) ?
- Use of Hep specific classes (4-vectors...)